

PRACTICAL-7

AIM: Create data frame using Pandas and perform basic data science operation.

OVERVIEW:

Pandas is a powerful Python library specifically designed for data analysis and manipulation. It offers high-performance, easy-to-use data structures called Series (one-dimensional arrays) and DataFrames (two-dimensional labeled data with columns).

Pandas excels in:

Loading data from various file formats (CSV, Excel, JSON, etc.)

Cleaning, transforming, and preparing data for analysis

Performing data aggregation and calculations

Time series analysis

Data visualization

Installation:

```
pip install pandas
```

Advantages:

- **High-Performance:** Pandas is built on top of NumPy, providing efficient memory usage and fast data operations.
- **Ease of Use:** The intuitive syntax makes data manipulation clear and readable.
- **Flexibility:** DataFrames can hold various data types (numerical, textual, booleans, etc.), making them versatile for different tasks.
- **Rich Ecosystem:** Pandas integrates seamlessly with other popular data science libraries like NumPy, Matplotlib, and Scikit-learn.
- **Extensive Functionality:** It offers a wide range of built-in functions for data cleaning, transformation, aggregation, and analysis.

Disadvantages:

- **Memory Usage:** While generally efficient, pandas can consume considerable memory for large datasets. Consider alternative libraries or data sampling techniques for extremely large datasets.
- **Learning Curve:** While simpler than some specialized libraries, pandas has a learning curve, especially for more advanced features.

- 3D Data Limitations: Pandas primarily focuses on two-dimensional data (DataFrames). For complex 3D data structures, consider using NumPy arrays.

CODE:**OUTPUT:**

```
import pandas as pd

data = [
    {'Name': 'Alice', 'Age': 25, 'City': 'New York'},
    {'Name': 'Bob', 'Age': 30, 'City': 'Los Angeles'},
    {'Name': 'Charlie', 'Age': 28, 'City': 'Chicago'}
]

df = pd.DataFrame(data)

print(df)
print(df.info())

ages = df['Age']
print(ages)

filtered_df = df[df['Age'] > 27]
print(filtered_df)

print(df['Age'].describe())

df['Country'] = 'USA'
```

OUTPUT:

```

|      Name  Age      City
0    Alice   25    New York
1      Bob   30  Los Angeles
2  Charlie   28     Chicago
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0    Name    3 non-null    object
1    Age      3 non-null    int64
2    City     3 non-null    object
dtypes: int64(1), object(2)
memory usage: 204.0+ bytes
None
0      25
1      30
2      28
Name: Age, dtype: int64
|      Name  Age      City
1      Bob   30  Los Angeles
2  Charlie   28     Chicago
count      3.000000
mean       27.666667
std        2.516611
min        25.000000
25%        26.500000
50%        28.000000
75%        29.000000
max        30.000000

```

CONCLUSION:

From this practical, I learn about the pandas installation and the basic operations of it.